



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,374	02/23/2004	Pal Takacsi-Nagy	BEAS-01389US2	8929
23910	7590	08/05/2008	EXAMINER	
FLIESLER MEYER LLP 650 CALIFORNIA STREET 14TH FLOOR SAN FRANCISCO, CA 94108			WANG, JUE S	
			ART UNIT	PAPER NUMBER
			2193	
			MAIL DATE	DELIVERY MODE
			08/05/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/784,374	<b>Applicant(s)</b> TAKACSI-NAGY ET AL.	
	<b>Examiner</b> JUE S. WANG	<b>Art Unit</b> 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 09 June 2008.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-3, 15-20 and 32-34 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-3, 15-20 and 32-34 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 29 October 2007 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)            | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>5/6/2008</u> .  | 6) <input type="checkbox"/> Other: _____                          |

**DETAILED ACTION**

1. Claims 1-3, 15-20 and 32-34 have been examined.
2. Claims 4-14, 21-31 and 35-42 were cancelled in Amendment dated 6/9/2008.

***Claim Rejections - 35 USC § 112***

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-13, 15-20 and 32-34 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following claim language is indefinite and not clear:

- i. As per claim 1, line 14, claim 18, line 13, the phrase “using a workflow program according to the workflow definition” is used. This phrase is not clearly understood because it is not clear if the workflow program contains workflow definitions that are specified in the form of annotation, or the workflow program contains workflow definitions that are specified by the language constructs defined by a second language and added to the first language.

Appropriate corrections are required.

Any claim not specifically addressed, above, is being rejected as incorporating the deficiencies of a claim upon which it depends.

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 1-3 and 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al., “Extending Java for High-Level Web Service Construction” (hereinafter Christensen), in view of Meredith (US 6,516,322 B1), further in view of Peltz, “Web Services Orchestration”.

7. As per claim 1, Christensen teaches the invention as claimed, including a method of extending an existing programming language, comprising the steps of: extending the existing language construct by adding at least one language construct defined by a second language (see page 6, paragraphs 3, 6, page 7, Fig 3, page 9; EN: element tags are XML constructs added to Java).

Christensen does not teach that the source code is extended with a plurality of workflow constructs defined by the second language, including adding an action construct representing an activity that allows a first software component written using the extended existing programming language to call an operation on a second software component written using the existing object oriented programming language; and using a workflow program according to the workflow definition, including processing, using a computer including a processing device operating thereon, the action construct to

Art Unit: 2193

allow the first software component written using the extended existing programming language to call the operation on the second software component written using the existing object oriented programming language.

Meredith teaches a workflow language SLANG which is written in XML (see column 12, lines 45-64), including an action construct representing an activity that allows a first software component written using SLANG to call an operation on a second software component written using another language, and using a workflow program according to the workflow definition, including processing the action construct to allow the first software component written using SLANG to call a second software component written using another language (see Figs 2a, 2b, 11a, 11b, column 13, lines 3-5, 46-55, column 14, lines 23-27).

It would have been obvious to one of ordinary skill in the art at the time of the invention that Christensen could be modified to integrate workflow constructs of the language SLANG into Java including the action construct because the approach of Christensen generalizes in a straightforward manner to any arbitrary interaction language described by an XML schema (see page 3, paragraph 4 of Christensen) and SLANG is an language described by an XML schema (see column 2, lines 17-37 of Meredith), and it would have been desirable to integrate the constructs of the SLANG language into Java because the XML constructs of SLANG provide compositional specification of autonomously executing systems such that the cost of developing and managing applications that span business units connected by communications networks is greatly reduced (see column 6, lines 26-55 of Meredith). Furthermore, it would have been obvious to one of ordinary skill in the art at the time of the invention that the first

Art Unit: 2193

software component (i.e., the descriptions of actions in terms of a port and a message written in SLANG, see column 13, lines 45-50 of Meredith) would be written using the extended existing programming language in the system of Christensen modified by Meredith (i.e., Java extended with SLANG) and the second software component called by the first software component (i.e., the common object model objects or other native technology behavior which are invoked as a result of mapping the action, see column 13, lines 3-5 of Meredith) would be written in the existing language in the system of Christensen modified by Meredith (i.e., Java) since Java is a popular programming language in the art with the advantage of being platform independent.

Christensen and Meredith do not teach program source file including a source code and classes therein and a workflow definition created using the existing object oriented programming language that is specified in the form of annotations to the source code and the classes.

Peltz teaches program source file including a source code and classes written in Java including a workflow definition specified in the form of annotations to the source code and the classes (see page 10, section BEA WebLogic Workshop).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Christensen and Meredith to include including a source code and classes written in Java including a workflow definition specified in the form of annotations to the source code and the classes as taught by Peltz because it allows developers to use the tags to specify web services conversations (see page 10, section BEA WebLogic Workshop, paragraphs 1, 3 of Peltz).

Art Unit: 2193

8. As per claim 2, Christensen further teaches that the second language is a markup language (see page 6, paragraph 3, page 9).

9. As per claim 3, Christensen further teaches that the second language is XML (see page 6, paragraph 3, page 9).

10. As per claims 18-20, these are the computer system claims of claims 1-3.

Therefore, they are rejected using the same reasons as claims 1-3.

11. Claims 15-17 and 32-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al., “Extending Java for High-Level Web Service Construction” (hereinafter Christensen), in view of Meredith (US 6,516,322 B1), further in view of Peltz, “Web Services Orchestration”, further in view of van der Aalst et al., “XML Based Schema Definition for Inter-Organization Workflow”, (hereinafter Aalst).

12. As per claim 15, Christensen, Meredith, and Peltz do not teach that said language construct is a multiple receive construct that allows a software component written using the extended existing programming language to wait on multiple input events received.

Aalst teaches the language XRL which is based on XML syntax and provides support for routing of workflow among trading partners for internet based electronic commerce services (see page 1, section 1, paragraph 1), where XRL has a multiple receive construct that allows a software component to wait on multiple input events received (the wait\_any construct, see pages 16-17, section 4.7).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Christensen, Meredith, and Peltz such that the XML integrated into Java contains constructs to specify workflow including a multiple receive construct that allows a software component written using the extended existing programming language to wait on multiple input events received as taught by Aalst because the approach of Christensen generalizes in a straightforward manner to any arbitrary interaction language described by an XML schema (see page 3, paragraph 4 of Christensen) and the XML constructs provide support for routing of workflow among trading partners to Internet based electronic commerce services to facilitate increased productivity and interoperability (see page 1, section 1, paragraph 1 of Aalst).

13. As per claim 16, Christensen, Meredith, and Peltz do not teach that said multiple receive construct further allows said software component proceed on a particular branch of program execution, based on the input event that occurred first within the said multiple input events.

While Aalst does not teach that the multiple receive construct allows software component proceed on a particular branch of program execution, based on the input event that occurred first within the said multiple input events. However, it would have been obvious that this functionality can be achieved in conjunction with the condition construct (see page 16, section 4.6, page 24, paragraph 2, and page 26, Figure 12) by placing the condition construct directly after the wait\_any construct such that different branches are taken depending on the event that was received by wait\_any. Furthermore, while Christensen, Meredith, Peltz and Aalst do not teach the multiple receive construct



Art Unit: 2193

has the specified functionality, it would have been obvious to one of ordinary skill in the art at the time of the invention to augment the multiple receive construct of XRL with such a functionality implemented since XML itself is an extensible language.

14. As per claim 17, Christensen Meredith, and Peltz do not teach that said construct is a looping construct with ordering of messages received, representing looping functionality, wherein the order allows said messages to be received in an order.

Aalst teaches the language XRL which is based on XML syntax and provides support for routing of workflow among trading partners for internet based electronic commerce services (see page 1, section 1, paragraph 1), where XRL has a looping construct representing looping functionality (see pages 16-17, section 4.7, the while\_do construct). While Aalst does not specifically teach that the loop construct has an ordering of messages received, where the ordering allows said messages to be received in an order, it would have been obvious that an ordering of the messages received does not need to rely on a looping construct, and instead can be achieved from the content of the loop. For example, in the mail order processing example presented on pages 24 –27, section 6, the while\_do loop is used to find a shipper where a list of shippers is contacted in order to find an available shipper. Since a message is received to determine whether the shipper can perform the shipment and the shippers are contacted in order, the messages are received in order because the shippers are contacted in order according to a list (see page 24, paragraph 2). Furthermore, while Aalst does not teach the loop construct has the specified functionality, it would have been obvious to one of ordinary skill in the art at

Art Unit: 2193

the time of the invention to augment the loop of XRL with such a functionality since XML itself is an extensible language.

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Christensen Meredith, and Peltz such that the XML integrated into Java contains constructs to specify workflow including a looping construct with ordering of messages received, representing looping functionality, wherein the order allows said messages to be received in an order as taught by Aalst because the approach of Christensen generalizes in a straightforward manner to any arbitrary interaction language described by an XML schema (see page 3, paragraph 4 of Christensen) and the XML constructs provide support for routing of workflow among trading partners to Internet based electronic commerce services to facilitate increased productivity and interoperability (see page 1, section 1, paragraph 1 of Aalst).

15. As per claims 32-34, these are the computer system claims of claims 15-17. Therefore, they are rejected using the same reasons as claims 15-17.

### ***Response to Arguments***

16. Rejection of claims 1 and 18 under §103(a):

Applicants argued that Christensen and Meredith do not teach the details of the extending step of claim 1 as amended which recites “extending an existing programming language, comprising the steps of: extending the existing language construct by adding at least one language construct defined by a second language so that the source code is extended with a plurality of workflow constructs defined by the second language,

Art Unit: 2193

including adding an action construct representing an activity that allows a first software component written using the extended existing programming language to call an operation on a second software component written using the existing object oriented programming language.” Applicants’ arguments have been fully considered and Examiner respectfully disagrees. Christensen teaches a method of extending an existing programming language by extending the existing language construct by adding at least one language construct defined by a second language (see page 6, paragraphs 3, 6, page 7, Fig 3, page 9; EN: element tags are XML constructs added to Java). Christensen does not specify any workflow constructs as part of the language constructs that are defined by the second language, however, Christensen does state that the method of extending generalizes in a straightforward manner to any arbitrary interaction language described by an XML schema (see page 3, paragraph 4 of Christensen). Therefore, it would have been obvious to one of ordinary skill in the art that the method of extending Java by XML constructs as taught by Christensen could have been modified to include workflow constructs in XML as taught by Meredith which includes an action construct that allows a first software component written using SLANG to call an operation on a second software component written using another language (see Figs 11a, 11b, column 13, lines 3-5, 46-55, column 14, lines 23-27) because SLANG is an language described by an XML schema (see column 2, lines 17-37 of Meredith), and it would have been desirable to integrate the constructs of the SLANG language into Java because the XML constructs of SLANG provide compositional specification of autonomously executing systems such that the cost of developing and managing applications that span business units connected by communications networks is greatly reduced (see column 6, lines 26-55 of Meredith).

Art Unit: 2193

Furthermore, it would have been obvious to one of ordinary skill in the art at the time of the invention that the first software component (i.e., the descriptions of actions in terms of a port and a message written in SLANG, see column 13, lines 45-50 of Meredith) would be written using the extended existing programming language in the system of Christensen modified by Meredith (i.e., Java extended with SLANG) and the second software component called by the first software component (i.e., the common object model objects or other native technology behavior which are invoked as a result of mapping the action, see column 13, lines 3-5 of Meredith) would be written in the existing language in the system of Christensen modified by Meredith (i.e., Java) since Java is a popular programming language in the art with the advantage of being platform independent.

### ***Conclusion***

17. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Bau III et al. (US 2003/0005181 A1) is cited to teach an annotation based development platform for asynchronous web services.

18. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

Art Unit: 2193

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2193

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./  
Supervisory Patent Examiner, Art Unit 2193

Jue Wang  
Examiner  
Art Unit 2193